



Project Name: Yoshi Test Report

Version 1.2

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Revision History

Date	Version	Description	Author
2015-01-11	1.0	Initial Draft	Yuxing Chen
2015-01-12	1.1	Revised grammar mistakes, check the abbreviations and add table names etc.	Martin anev and Yuxing Chen
2015-01-13	1.2	Small edits, adding the list of figures and tables	Martin Anev

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Table of Contents

LIST OF TABLES	3
LIST OF FIGURES.....	3
1. INTRODUCTION.....	4
1.1 PURPOSE OF THIS DOCUMENT	4
1.2 DOCUMENT ORGANIZATION	4
1.3 INTENDED AUDIENCE.....	4
1.4 DEFINITIONS AND ACRONYMS.....	4
1.4.1 <i>Definitions</i>	4
1.4.2 <i>Acronyms and abbreviations</i>	4
2. PROJECT SPECIFIC DOCUMENT REFERENCE	5
3. FEATURES TO BE TESTED	6
3.1 PASS/FAIL CRITERIA	6
3.2 TEST SPECIFICATIONS AND PROCEDURES.....	6
4. TEST ALLOCATION OF REQUIREMENT.....	12

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

List of Tables

Table 1. Definitions	4
Table 2. Acronyms and abbreviations	4
Table 3. Abbreviation of Documents.....	5
Table 4. Severity Rankings for Discrepancies.....	6
Table 5. Test of Wrong Repository Name Input.....	6
Table 6. Test of Correct Repository Name Input.....	7
Table 7. Test of Community Type Evaluation.....	7
Table 8. Test of Visual Evaluation of Community	8
Table 9. Test of Number of Community Type 1	10
Table 10. Test of Number of Community Type 2	11
Table 11. The Test to validate Requirement.....	12

List of Figures

Figure 1. Test result of T1.1	6
Figure 2. Test result of T1.2	7
Figure 3. Text result of T2.1	8
Figure 4. Community Decesion tree	9
Figure 5. The test result of T2.2 Part 1	9
Figure 6. The test result of T2.2 Part 2	10

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

1. Introduction

1.1 Purpose of this document

The purpose of this document is to give a test report according to acceptance test plan delivered by Team Yoshi in the Distributed Software Development done simultaneously in ‘Politecnico di Milano’ situated in Milan, Italy and ‘Mälardalen University’ situated in Västerås, Sweden.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*
- Section 2, *Project specific document reference*
- Section 3, *Features to be tested*
- Section 4, *The allocation of requirements*

1.3 Intended Audience

The intended audience is:

- The customer of the project
- The supervisors of the project
- Yoshi Team
- All related stakeholders
- Any developer with interest to continue or improve the project

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Community	Social unit of any size that shares common values.
Acceptance Test	A test conducted to determine if the requirements have met
Compute a Community	The action of observing a community, gaining definitive metrics for Social Communities and computing these metrics. The outcome of the action is a decision – what type is the observed community.
Visualize a community	The action of observing the output of the decision “What type is a community?”. The deliverable is text and images.
Community Decision Tree	Decision tree defining what type is an observed social community based on some characteristics. The Decision Tree is discussed in the document ‘Uncovering Latent Social Communities in Software Development’ [4].

Table 1. Definitions

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
FR	Functional Requirement
JSON	Java Script Objected Notation

Table 2. Acronyms and abbreviations

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

2. Project Specific Document Reference

Abbreviations	Document name and version
[PRD]	Project Requirements document, Yoshi team, MDH/Pol. v 3.2.1
[DD]	Design Description Document v3
[PPD]	Project Plan document v1.1
[ATP]	Acceptance Test Plan v1.0

Table 3. Abbreviation of Documents

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

3. Features to be tested

The Yoshi Vis system will adhere to the required requirements and will test the required features and design of Yoshi Vis. The features of the software can be found in [DD].

3.1 Pass/Fail Criteria

Severity	Description
Critical	Discrepancies that halt further program execution. Example: run-time errors that cause the system to lock up in an unexplained or unexpected manner.
Major	Discrepancies that cause the application not to perform as functionally required. Example: inability to print a report.
Minor	Discrepancies that are not considered critical or major. Examples: misspellings on a screen, ambiguous Help messages.

Table 4. Severity Rankings for Discrepancies

3.2 Test Specifications and Procedures

The following tests are strictly according to [ATP].

Test ID	T1.1
Test Name	Repository name input
Description	Enter the wrong name of a repository to see if it shows an error message
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "anriod" in the input field of "Repository name" • Click "OK"
Output Specification	"Unable to find community anriod"
Pass/Fail	Pass

Table 5. Test of Wrong Repository Name Input

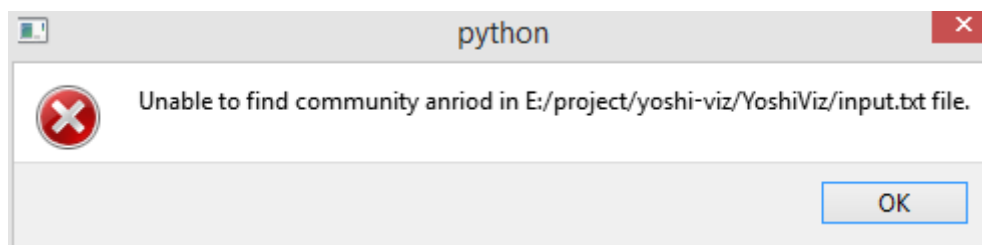


Figure 1. Test result of T1.1

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Test ID	T1.2
Test Name	Repository name input
Description	Enter the correct name of a repository to see if it shows the result
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "android" in the input field of "Repository name" • Click "OK"
Output Specification	"The type of android is Informal Network, Informal Community. Check output for more information"
Pass/Fail	pass

Table 6. Test of Correct Repository Name Input

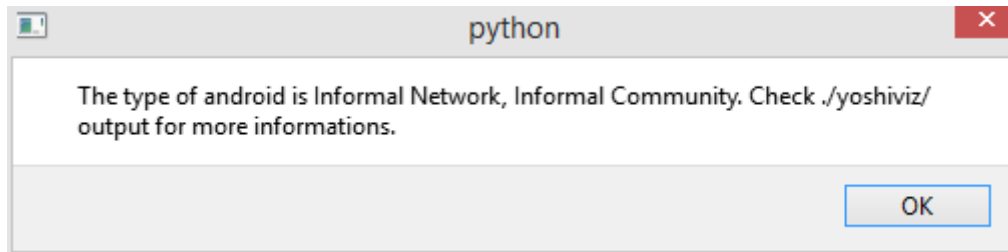


Figure 2. Test result of T1.2

Test ID	T2.1
Test Name	Community type evaluation
Description	Enter the correct name of a repository to see if it shows the required result, i.e. the type of community.
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "android" in the input field of "Repository name" • Click "OK"
Output Specification	"The type of android is Informal Network, Informal Community. Check output for more information"
Pass/Fail	Pass

Table 7. Test of Community Type Evaluation

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Yoshi-Viz

android

This community is Informal Network, Informal Community.

This community type has been found using the following properties.

Quality related attributes:

- This community has a wiki page.
- Reopened issues percentage: 0.11
- Average commits per user: 262.5
- Average number of files modified by contributor: 1.236
- Average subscriptions: 34.358
- Average clustering coefficient: 0.589
- Degree centrality: 16.474
- Modularity: 0.235
- Closeness centrality: 1.915

Community related attributes:

- Average period for a milestone accomplishment: 0.008
- Hierarchy degree: 0.576
- Average project collaboration: 34.153
- Percentage of profesional contributors: 0.245
- Average physical distance among contributors: 6683.196
- Average cultural distance: 22.03
- Self similarity percentage: 0.728

Figure 3. Text result of T2.1

Test ID	T2.2
Test Name	Visual evaluation of the community
Description	Enter the correct name of a repository to see if it shows the result in a visual form
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "android" in the input field of "Repository name" • Click "OK"
Output Specification	<p>"The type of android is Informal Network, Informal Community. Check output for more information"</p> <p>Check if the visualized result(decision tree) in output folder</p>
Pass/Fail	Pass

Table 8. Test of Visual Evaluation of Community

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Decision Tree

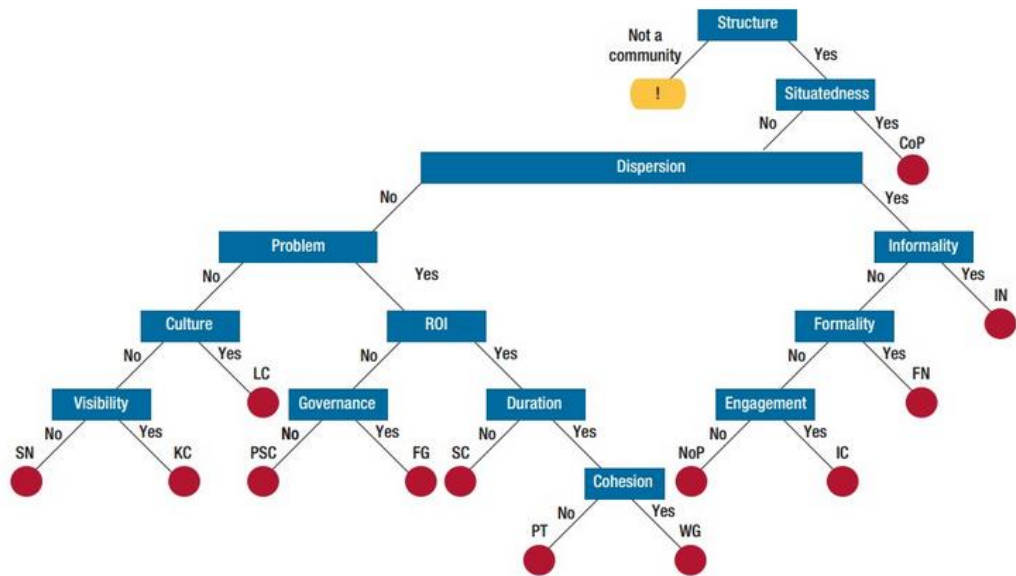


Figure 4. Community Decision tree

Decision Tree Cuts

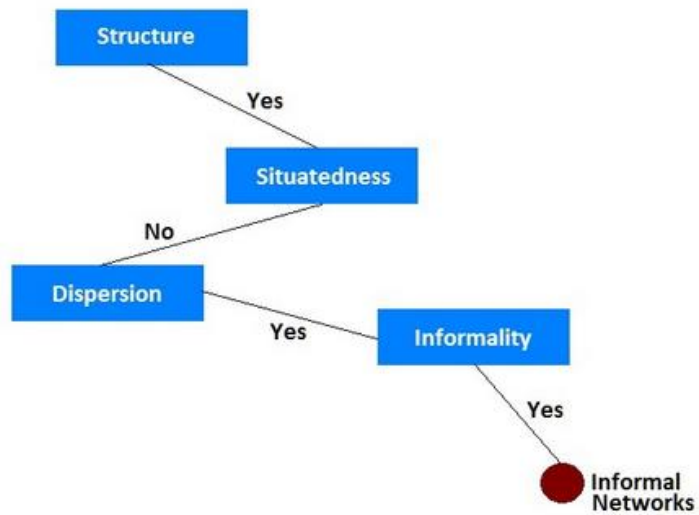


Figure 5. The test result of T2.2 Part 1

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

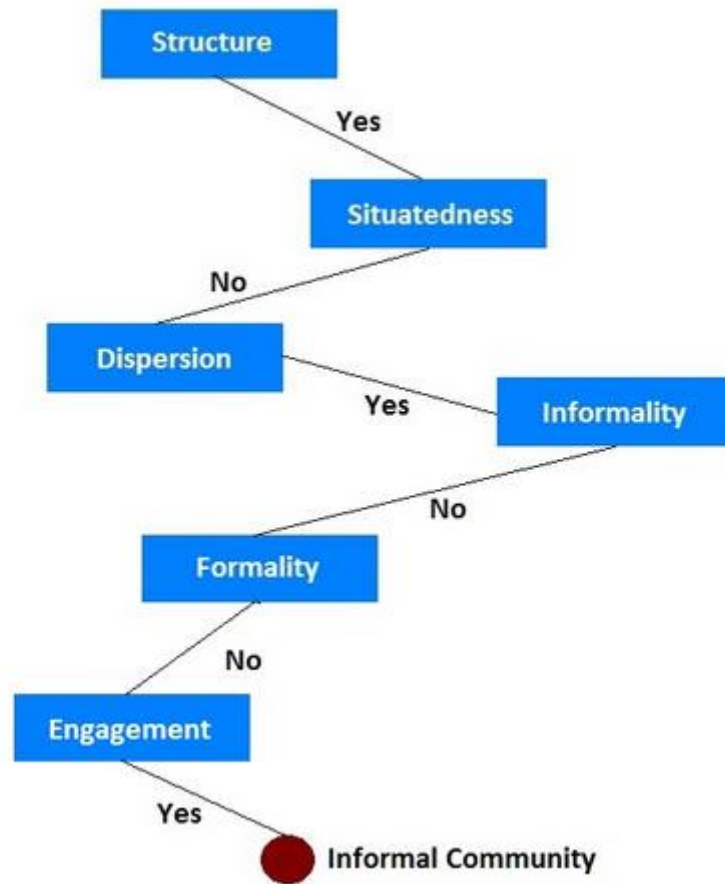


Figure 6. The test result of T2.2 Part 2

Test ID	T3.1
Test Name	Number of community type(s)
Description	Enter the correct name of a repository to see if it shows the result where the repository can belong to more than one community type.
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "android" in the input field of "Repository name" • Click "OK"
Output Specification	"The type of android is Informal Network, Informal Community. Check output for more information"
Pass/Fail	Pass

Table 9. Test of Number of Community Type 1

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

Test ID	T3.2
Test Name	Number of community type(s)
Description	Enter the correct name of a repository to see if it shows the result where the repository can belong to only one type of community.
Pre-requisite(s)	The software is installed and has been executed
Input Specification	<ul style="list-style-type: none"> • Open the file "main.py" in the main directory. • Click "Browse" to load the json data file • Type "MonoGame" in the input field of "Repository name" • Click "OK"
Output Specification	"The type of MonoGame is Formal Network. Check output for more information"
Pass/Fail	Pass

Table 10. Test of Number of Community Type 2

Yoshi	Version: 1.2
Test Report	Date: 2015-01-13

4. Test Allocation of Requirement

FR id	FR Description	Test ID
FR1	The current output of Yoshi should be prepared for adaptation	Every test used the Yoshi output which we made into JSON file
FR2	provide proper adapter for the formatted output of Yoshi.	Every test used the Yoshi output which we made into JSON file
FR3	take decision what type is an observed social community	T2.1
FR4	visualize as output the formatted output of Yoshi.	T2.2
FR5	visualize as output the evaluated decision of the type in text form.	T3.1 T3.2
FR6	visualize as output motivation of the taken decision.	T2.1 T2.2 T3.1 T.3 The result is in output file folder
FR7	visualize as output the type on the Community Decision Tree in graphical form.	T2.1 T2.2 T3.1 T.3 The result is in output file folder

Table 11. The Test to validate Requirement

As shown above, the corresponding functional requirement and tests are fulfilled. All requirements in this section are satisfied by the Tests in Section 3.